浙江大学
ZHEJIANG UNIVERSITY

Computer Architecture Experiment

Topic 7. Pipelined CPU accessing Mem. in multiple cycle

浙江大学计算机学院
陈文智
chenwz@zju.edu.cn

实验操作流程

- 阅读实验文档，理解存储器的等待周期对性能的影响
- 以前一次实验为基础，修改Memory访问机制为多个时钟周期访问时间，CP0增加CPU Tick计算功能，并测试实际循环指令执行效果，分析实际存储器对CPU性能的影响。
- 对处理器进行仿真，检验处理器的仿真结果是否符合要求。
- 综合工程并下载至开发板，在单步执行的过程中检查调试屏幕的输出，检验处理器的执行过程是否正确。

实验验收标准

- 仿真执行过程中，处理器的行为和内部控制信号均符合要求。
- 下载至开发板后的单步执行过程中，寄存器的变化过程和最终执行结果与测试程序相吻合。

Outline

- Experiment Purpose
- Experiment Task
- Basic Principle
- Operating Procedures
- Precaution
- Checkpoints

Experiment Purpose

- Understand the principle of CPU accessing Mem. in multiple cycle and it's affection to CPU.
- Master the design methods of pipelined CPU accessing Mem. in multiple cycle.
- master methods of program verification of Pipelined CPU accessing Mem. in multiple cycle.

Experiment Task

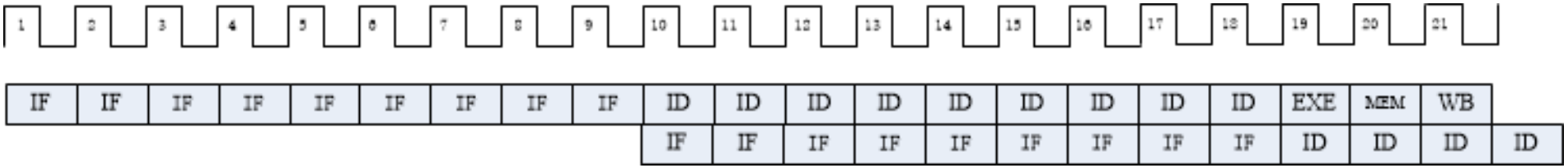
- **Design of Pipelined CPU accessing Mem. in multiple cycle**
  - Redesign Inst. ROM & Data RAM
  - Modify CPU Controller
  - Modify datapath

- **Verify the Pipelined CPU with program and observe the execution of program**

Access Mem. in multiple cycles

- While accessing Mem., 8 cycles for addressing, get value and delivery(or addressing, delivery and put value).
- Another cycle for reset signal “ACK”(used for cache)
- Above all, for accessing mem., it needs 9 cycle, and stalls the pipeline in 8 cycles.

Execution of CPU with Mem. accessed in multiple cycles



Inst. ROM's Job

- **Initial State: S_IDLE**
- **Working State: S_READ**
- **Read for 7 cycles, then output the value, set the signal ACK**
- **STALL = CS & ~ACK**

Data RAM's Job

- **Initial State: S_IDLE**
- **Working State: S_READ/S_WRITE (according to signal we)**
- **Read/Write for 7 cycles, then output the value/write the value, set the signal ACK**
- **STALL = CS & ~ACK**

When ROM&ROM STALL

- **Inst. ROM Stalls, Stall should be in ID stage otherwise jump indicator would be lost**
  - if_en=0, id_en=0, rst_exe=1
- **DATA RAM stalls**
  - …, rst_wb=1

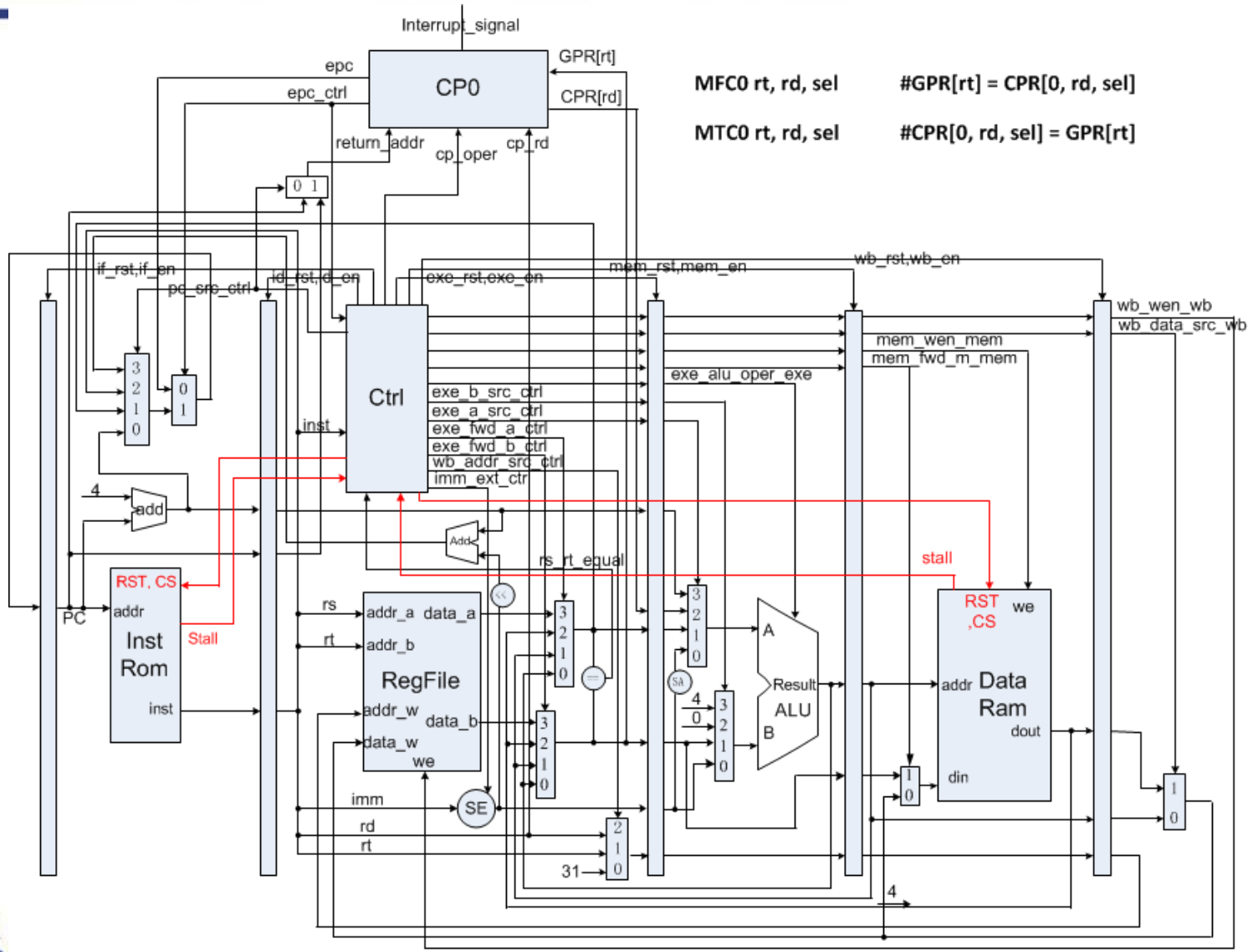
Add TCR in CP0 (1)

- TCR means “Tick Counter Register”
- TCR increments while CPU runs in cycles
- Program could get TCR via MFC0 instr.
- In localparams: CP0_TCR = 9;

Add TCR in CP0 (2)

```
// CP0 registers
localparam
        //CP0_SR    = 0,
        //CP0_EAR   = 1,
        CP0_EPCR  = 2,
        CP0_EHBR  = 3,
        //CP0_IER   = 4,
        //CP0_ICR   = 5,
        //CP0_PDBR  = 6,
        //CP0_TIR   = 7,
        //CP0_WDR   = 8,
        CP0_TCR   = 9;
```

Datapath of CPU accessing Mem. in multiple cycle



Code Example (1)

```
module inst_rom (
        input wire clk,
        input wire rst,
        input wire cs,
        input wire [31:0] addr,
        output reg [31:0] dout,
        output reg stall
        );
```

Code Example (2)

```
module data_ram (
        input wire clk,
        input wire rst,
        input wire cs,
        input wire we,
        input wire [31:0] addr,
        input wire [31:0] din,
        output reg [31:0] dout,
        output reg stall
        );
```

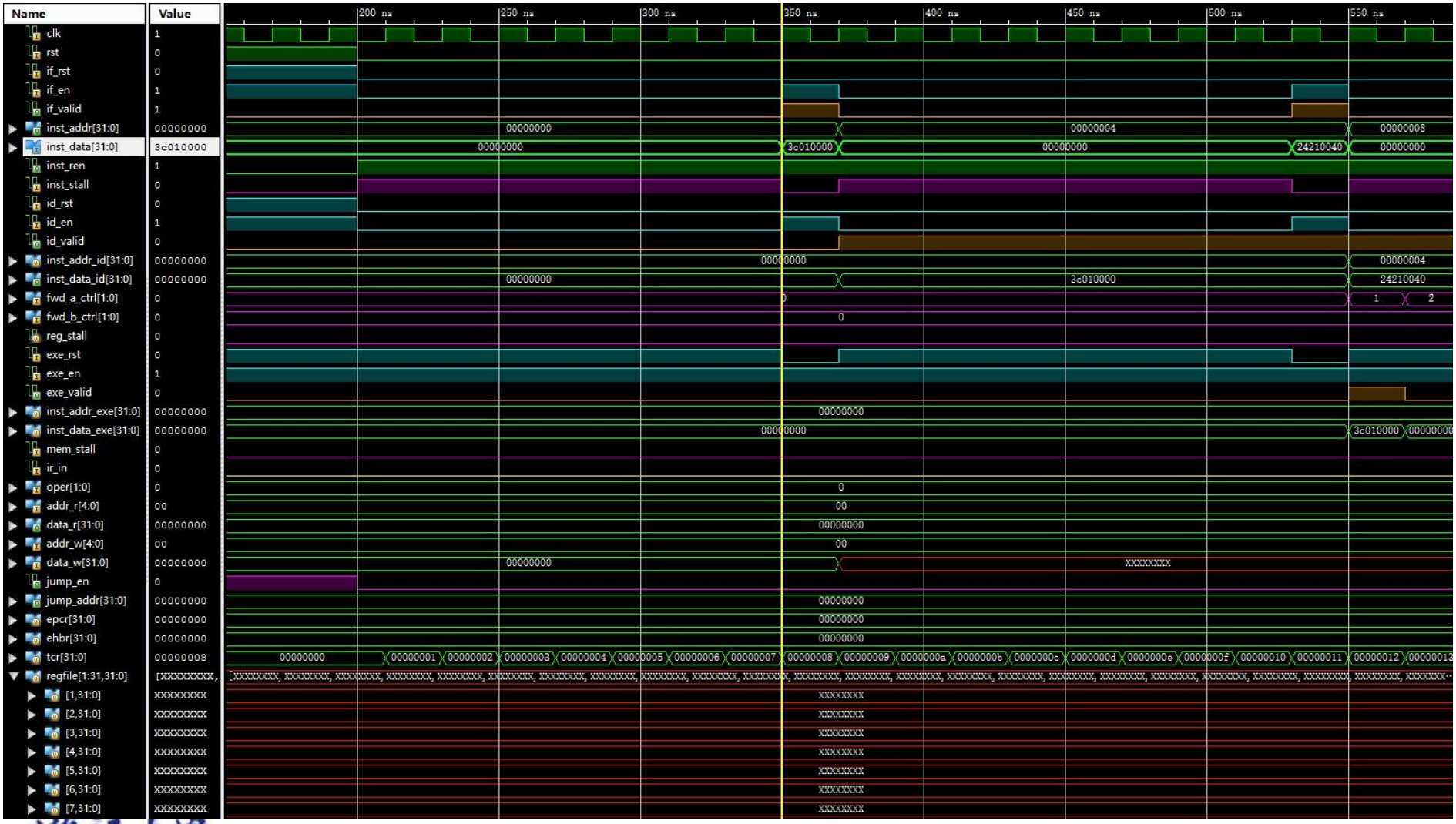
Instr. Mem.(1)

```
0:    3c010000    lui    R1,0x0        //main entry
4:    24210040    addiu  R1,R1,64
8:    40811800    mtc0   R1, R3
c:    00001020    add    R2,R0,R0
10:   00001820    add    R3,R0,R0
14:   40044800    mfc0   R4,R9
18:   00002820    add    R5,R0,$0
1c:   20420001    addi   R2,R2,1       //loop
20:   28412710    slti   R1,R2,10000
24:   1420fffd    bne    $1,R0,-2      //jump to loop
28:   00000000    nop
```

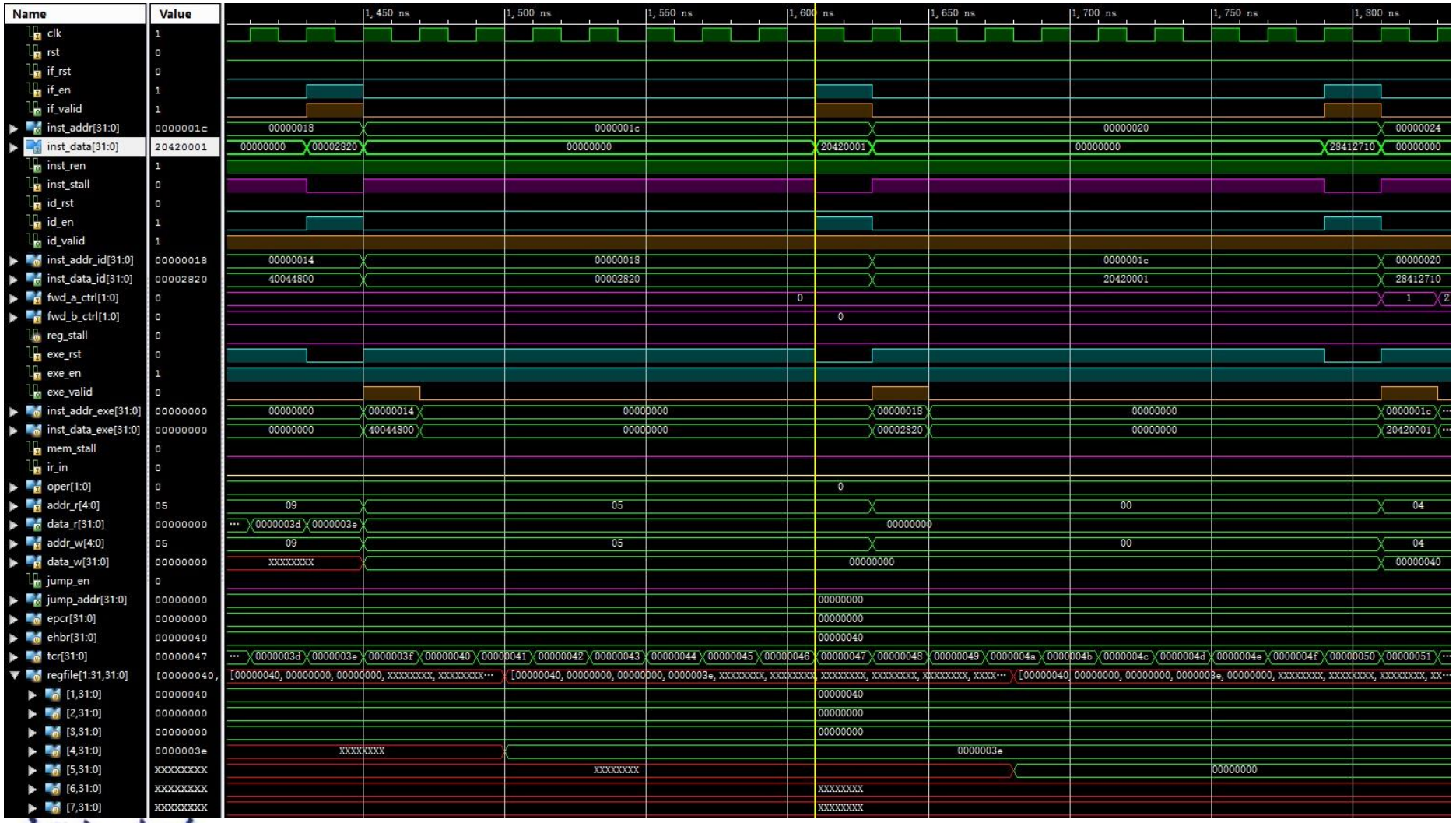
Instr. Mem.(2)

```
2c:     40054800    mfc0    R5, R9
30:     00a42822    sub     R5, R5, R4
34:     20420001    addi    R2,R2,1         //loop
38:     0800000d    j       34
3c:     00000000    nop
40:     20630001    addi    R3,R3,1         //handler
44:     42000018    eret
48:     00000000    nop
```

Simulation (1)



Simulation (2)



Checkpoints

- **CP 1:**

  Waveform Simulation of the Pipelined CPU with the verification program

- **CP 2:**

  FPGA Implementation of the Pipelined CPU with the verification program

Thanks!

浙江大学
ZheJiang University